

## 1 Strong Duality Theorem<sup>1</sup>

Throughout this task, Let  $K \in \{\mathbb{Q}, \mathbb{R}\}$ ,  $A \in K^{m \times n}$ ,  $b \in K^{1 \times m}$ . We consider the primal and dual problem in the following form:

$$P = \max_{Ax \leq b} c^T x \qquad D = \min_{\substack{y^T A = c^T \\ y \geq 0}} y^T b,$$

which are easily transformed into the forms considered in the lecture.

We note that the part of the Strong Duality Theorem that claims that unboundedness resp. infeasibility of the primal implies infeasibility resp. unboundedness of the dual and vice versa is partly wrong and partly readily proved:

We do not need to consider the “vice versa” part, as this one can be reduced to the other case, since the dual of the dual is equivalent to the original problem. Suppose,  $y$  was a dual solution and the primal was unbounded. Then  $y^T b \geq y^T A x = c^T x$  is arbitrarily big, which cannot hold for any fixed  $y$ . Thus the dual is infeasible. On the other hand, the infeasibility of the primal does not imply the unboundedness of the dual as the classical example

$$\begin{aligned} \max \quad & 2x - y \\ \text{such that} \quad & x - y \leq 1 \\ & -x + y \leq -2 \\ & x, y \geq 0, \end{aligned}$$

which is equivalent to its dual and infeasible, shows. For the proof of the second part we use the following lemma:

**Lemma 1 (Farkas)**  $(\exists x: Ax \leq b) \Leftrightarrow (K^m \ni u \geq 0 \wedge u^T A = 0 \geq 0 \Rightarrow u^T b \geq 0)$

PROOF First, we show the direction “ $\Leftarrow$ ”. Let  $Ax \leq b$  be given. By multiplication of  $A$  and  $b$  with positive constants, we may assume, that the elements in  $A$ ’s first column are 0 or  $\pm 1$  and by reordering the rows of  $A$  (and the elements of  $b$ ) that it has the form

$$\begin{aligned} \forall 1 \leq i \leq m_1: \quad & a'_i x' \leq b'_i \\ \forall m_1 + 1 \leq i \leq m_2: \quad & x_1 + a'_i x' \leq b'_i \\ \forall m_2 + 1 \leq i \leq m: \quad & -x_1 + a'_i x' \leq b'_i, \end{aligned}$$

---

<sup>1</sup>“Kombinatorische Optimierung” by Jens Vygen and Bernhard Korte and lecture notes by Stephan Held from the University of Bonn were used.

where  $a'_i$  is the row vector that contains the second to  $n$ th element of  $A$ 's  $i$ th row in the multiplied system and  $b_i$  is the vector of the right hand sides of the multiplied system. This system of equations is equivalent (as regards solvability) to the system

$$\begin{aligned} \forall 1 \leq i \leq m_1: \quad & a'_i x' \leq b_i \\ \forall m_1 + 1 \leq j \leq m_2 \forall m_2 + 1 \leq k \leq m: \quad & a'_j x' - b'_j \leq b'_k - a'_k x', \end{aligned}$$

which follows, since inequality constraints that are not taken with equality may be omitted without changing the solvability.

Repeating this, we may reduce the dimension of  $x$  to zero. We observe, that the original system  $Ax \leq b$  is solvable if in this final system every right hand side scalar is nonnegative. But we observe, that in the final system the coefficients on the left hand side are just nonnegative linear combinations of the coefficients in the respective column in  $A$  that vanish, i.e.  $u^T A = 0$  for some nonnegative vector  $u$  that describes the linear combination. The right hand side of the system is then  $y^T b$ , which is nonnegative by our assumption. Thus,  $Ax \leq b$  is solvable.

For the other direction:  $Ax \leq b$  implies  $u^T Ax \leq u^T b$  for any  $u \geq 0$ , which implies  $0 \leq u^T b$  if in addition  $u^T A = 0$ . This finishes the proof.

**Theorem 1** *If  $P, D \in K$ , we have  $P = D$ .*

PROOF We have

$$\forall x \in \{x | Ax \leq b\} \forall y \in \{y | y^T A = c^T \wedge y \geq 0\}: c^T x = y^T Ax \leq y^T b,$$

therefore it remains to show, that

$$\exists x \in \{x | Ax \leq b\} \exists y \in \{y | y^T A = c^T \wedge y \geq 0\}: c^T x \geq y^T b$$

This is equivalent to the solvability of

$$\begin{aligned} Ax & \leq b \\ A^T y & = c \\ -c^T x + y^T b & \leq 0 \\ y & \geq 0 \end{aligned}$$

which is, by Farkas 1, equivalent to the infeasibility of

$$\begin{aligned} u^T A - wc^T & = 0 \\ v^T A^T + wb^T & \geq 0 \\ u^T b + v^T c & < 0 \\ u & \geq 0 \\ w & \geq 0 \end{aligned}$$

We suppose, this was not the case. If there was a solution of the form  $(u, v, 0)$ , we would have a solution to

$$\begin{aligned} u^T A &= 0 \\ v^T A^T &\geq 0 \\ u^T b + v^T c &< 0 \\ u, w &\geq 0 \end{aligned}$$

This is equivalent by Farkas 1 to the infeasibility of

$$Ax \leq b \qquad A^T y = c \qquad y \geq 0$$

which contradicts the feasibility of the dual.

If otherwise we had a solution of the form  $(u, v, w), w > 0$ . Then we may scale the whole system to have a solution  $(u, v, 1)$  and so

$$0 > u^T b + v^T c \geq u^T (-Av) + v^T (A^T u) = 0$$

which is a contradiction and concludes the proof.

## 2 Maximum Disjoint Paths

The dual of the given linear problem is :

$$\begin{aligned} \min \quad & \sum_{e \in E} y_e \\ \text{s.t.} \quad & \forall p \in P, \sum_{e \in p} y_e \geq 1 \\ & \forall e \in E, y_e \geq 0 \end{aligned}$$

Indeed, if we want an upper-bound on the value of the maximization, we have to take something that includes at least once every  $x_p$ , which means at least one of the edge that is contained by  $p$  must be taken into account.

This duality is simply the LP version of MAXFLOW/MINCUT duality when all weights are equal to 1. A binary solution of the above dual gives a cut of the graph with a minimum number of edges.

## 3 Maximum Weight Spanning Tree

We consider Kruskal's algorithm, the maximizing greedy algorithm on the graphical matroid. We want to prove the optimality of Kruskal's for not necessarily nonnegative edge weights using the LP on the exercise sheet. Obviously, the incidence vector  $x$  of the tree returned by the algorithm provides a primal solution.

To show optimality, it suffices to construct a solution to the dual problem  $y$  such that  $x, y$  satisfy complementary slackness.

As performed in the lecture, we construct a dual feasible solution using the tree structure  $T_K$  as discussed in the lecture and define a dual solution  $y^*$ . Using the notation from the lecture, we get.

If  $y_S^* > 0$ , then  $S = V(e_j)$

## 4 $k$ -Disjoint Perfect Matchings

The integrality of the bipartite perfect matching polytope gives us a polynomial time algorithm  $A$  to compute a perfect matching. Given this algorithm, and a  $k$ -regular bipartite graph  $G = (A \cup B, E)$ , we proceed as follows :

```
while k > 0
  new_matching := A(G)
  perfect_matching_partition.add(new_matching)
  E = E \ new_matching
  k--
endwhile
```

The correctness of this algorithm is easily verified by realising that the retrieval of edges from a perfect matching to a  $k$ -regular bipartite graph gives a  $(k - 1)$ -regular bipartite graph. It also clearly runs in polynomial time.